Weather Forecasting Application using Machine Learning

Code & Execution:

WeatherForecastingAppUsingMLipynb

File Edit View Insert Runtime Tools Help

```
from google.colab import drive
```

```
[ ] drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import graphviz
```

```
[ ] df = pd.read_csv('/content/drive/MyDrive/data/Weatherlink_UMIT_SNDTWU_6-30-23_12-00_AM.csv', encoding="ISO-8859-1")
```

```
[ ] df.head()
```

	Weatherlink UMIT SNDTWU	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 40	Unnamed: 41	Unnamed: 42	Unnamed: 43	Unnamed: 44	Unnamed: 45	Unnamed: 46	Unnamed: 47	Unnamed: 48	Unnamed: 49
0	6/30/23 12:00 AM : 1 Week	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU	weather link UMIT SNDTWU
2	NaN	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Inside Temp/Hum	Barometer	...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...	Vantage Pro2 Plus, includes UV & Solar Radiati...
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			High				High		Inside	Inside					High					Heating	Cooling

Connected to Python 3 Google Compute Engine backend (GPU)

The final data

```
df
```

	Date & Time	Barometer - in Hg	Absolute Pressure - in Hg	Temp - °C	Hum - %	Dew Point - °C	Avg Wind Speed - km/h	High Rain Rate - in/h	Solar Rad - W/m^2
1	6/30/23 12:00 AM	29.67	29.66	26	87	24	17	0.14	0
2	6/30/23 12:05 AM	29.67	29.65	26	86	24	16	0.06	0
3	6/30/23 12:10 AM	29.67	29.66	26	86	24	9	0.04	0
4	6/30/23 12:15 AM	29.67	29.66	26	87	24	13	0	0
5	6/30/23 12:20 AM	29.67	29.65	26	87	24	11	0	0
...	...	...	...	...	...	...	...	...	...
2013	07-06-2023 23:40	29.58	29.56	26	91	25	16	0.05	0
2014	07-06-2023 23:45	29.58	29.57	26	91	25	17	0.04	0
2015	07-06-2023 23:50	29.58	29.57	26	91	25	12	2.7	0
2016	07-06-2023 23:55	29.58	29.57	26	91	25	21	0.47	0
2017	07-07-2023 00:00	29.58	29.57	26	91	24	21	0.19	0

2017 rows × 9 columns

+ Code + Text

Exploring the Final Dataset

```
[ ] print('The shape of our features is:', df.shape)
```

```
The shape of our features is: (2017, 9)
```

```
df.describe()
```

	Date & Time	Barometer - in Hg	Absolute Pressure - in Hg	Temp - °C	Hum - %	Dew Point - °C	Avg Wind Speed - km/h	High Rain Rate - in/h	Solar Rad - W/m^2
count	2017	2017	2017	2017	2017	2017	2017	2017	2017
unique	2017	29	29	6	15	5	27	201	375
top	07-07-2023 00:00	29.61	29.52	27	92	25	8	0	0
freq	1	163	128	679	260	1029	189	1254	966

```
[ ] df.info()
```

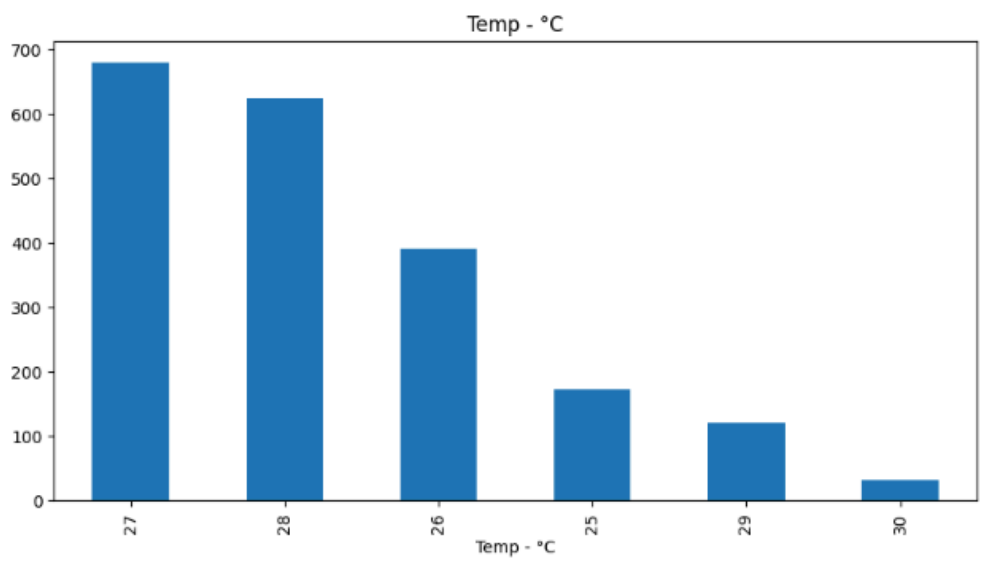
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2017 entries, 1 to 2017
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Date & Time                 2017 non-null   object
 1   Barometer - in Hg           2017 non-null   object
 2   Absolute Pressure - in Hg   2017 non-null   object
 3   Temp - °C                   2017 non-null   object
 4   Hum - %                     2017 non-null   object
 5   Dew Point - °C              2017 non-null   object
 6   Avg Wind Speed - km/h       2017 non-null   object
 7   High Rain Rate - in/h       2017 non-null   object
 8   Solar Rad - W/m^2           2017 non-null   object
dtypes: object(9)
memory usage: 141.9+ KB
```

```
[ ] # Converting object dataframes into numeric
```

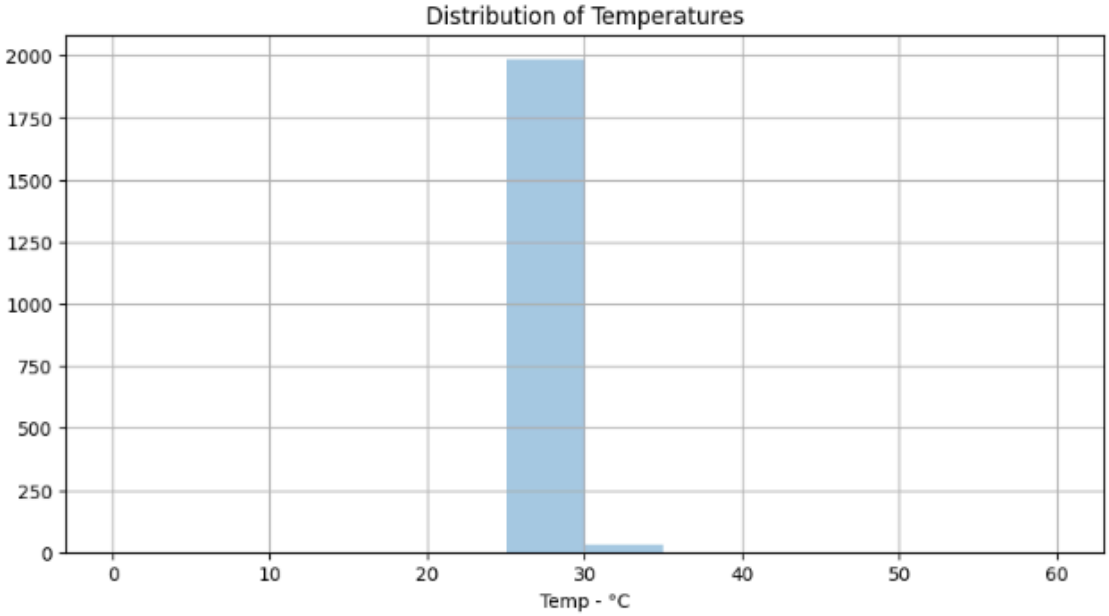
Quick Analysis of weather

+ Code + Text

```
[ ] plt.figure(figsize=(10,5))
    df['Temp - °C'].value_counts().head(15).plot(kind='bar')

    plt.title('Temp - °C')
    plt.show()
```



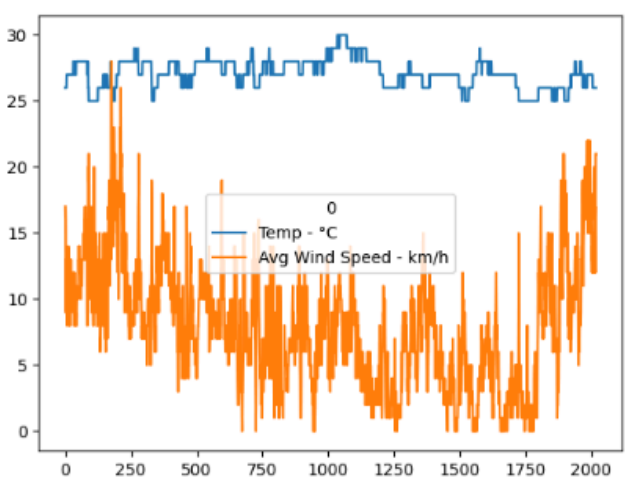
```
sns.distplot(df['Temp - °C'],bins=[i for i in range(0,61,5)], kde=False)
```



- Most common temperature scale is from 25 to 30 degree.

```
[ ] df[['Temp - °C',  'Avg Wind Speed - km/h']].plot()
```

```
<Axes: >
```



- This shows that these features has no gap in between i.e we are not missing any data

```
from tensorflow.keras.layers import Dense,RepeatVector, LSTM, Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import TimeDistributed
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D
from tensorflow.keras.models import Sequential
```

```
[ ] from tensorflow.keras.layers import Bidirectional, Dropout
```

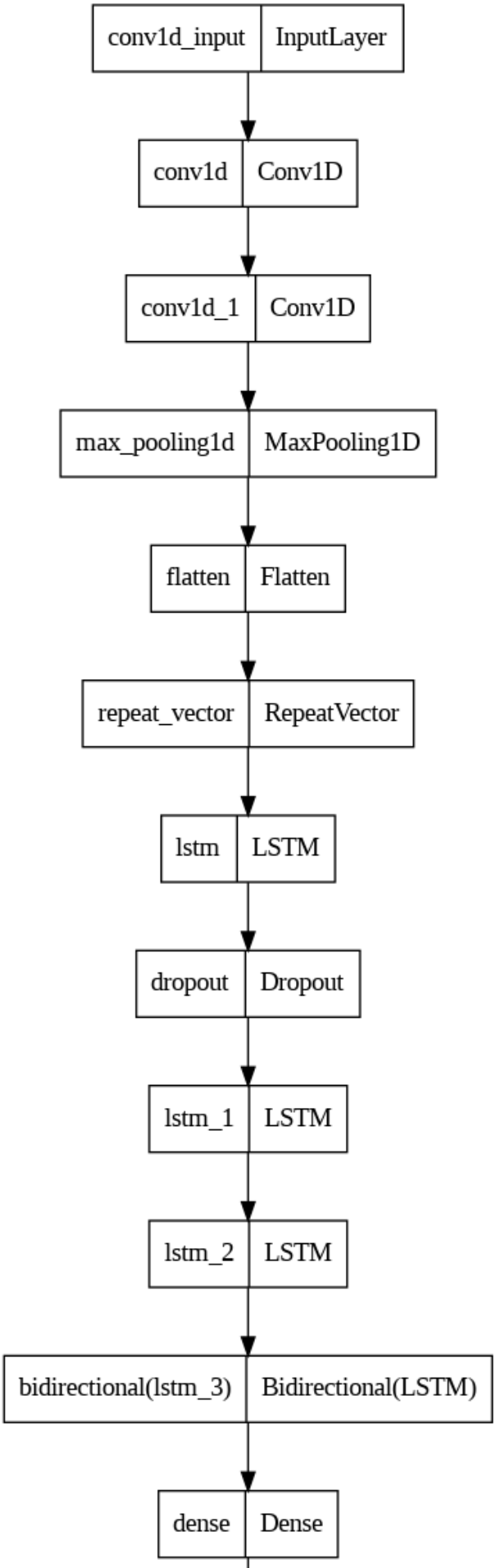
```
[ ] model = Sequential()
    model.add(Conv1D(filters=256, kernel_size=2, activation='relu', input_shape=(30,1)))
    model.add(Conv1D(filters=128, kernel_size=2, activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Flatten())
    model.add(RepeatVector(30))
    model.add(LSTM(units=100, return_sequences=True))
    model.add(Dropout(0.2))
    model.add(LSTM(units=100, return_sequences=True))
    model.add(LSTM(units=100, return_sequences=True))
    model.add(Bidirectional(LSTM(128, activation='relu')))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(1))
    model.compile(loss='mse', optimizer='adam')
    history = model.fit(Xtrain,Ytrain,epochs=300, verbose=1 )
```

```
WARNING:tensorflow:Layer lstm_3 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_3 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_3 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Epoch 1/300
44/44 [==============================] - 17s 95ms/step - loss: 0.0635
Epoch 2/300
```

```
[ ] model.save("model_trained_temp.h5")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserW
  saving_api.save_model(
```

```
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='model.png')
```


conv1d_input
InputLayer
conv1d
Conv1D
conv1d_1
Conv1D
max_pooling1d
MaxPooling1D
flatten
Flatten
repeat_vector
RepeatVector
lstm
LSTM
dropout
Dropout
lstm_1
LSTM
lstm_2
LSTM
bidirectional(lstm_3)
Bidirectional(LSTM)
dense
Dense

Temperature:

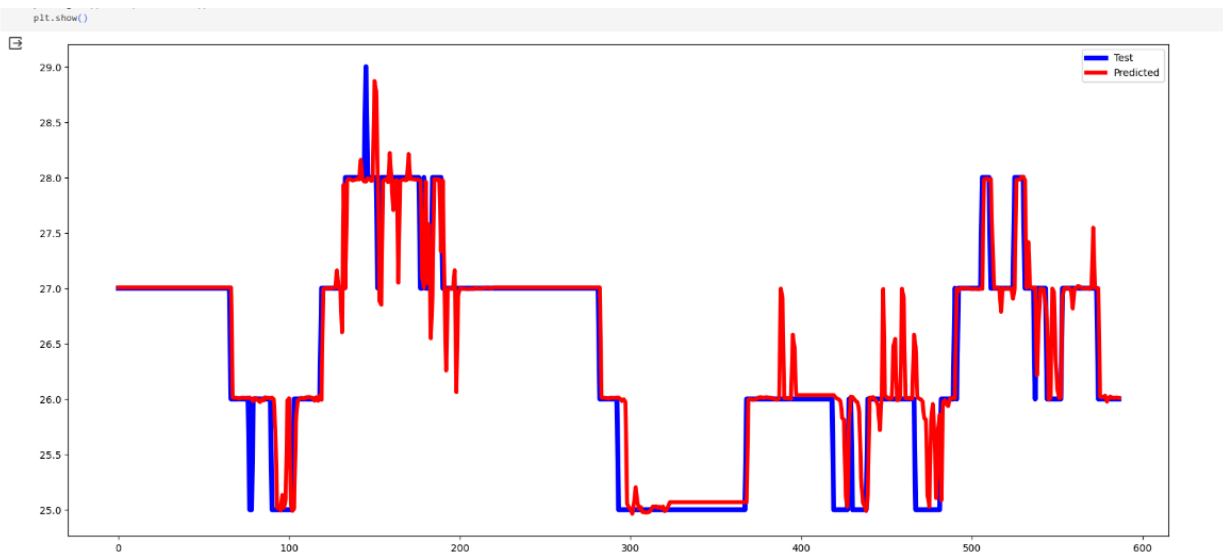
```
[ ] predict = model.predict(Xtest)
```

```
19/19 [==============================] - 2s 19ms/step
```

```
[ ] predict = scalar.inverse_transform(predict)
```

```
[ ] Ytesting = scalar.inverse_transform(Ytest)
```

```
plt.figure(figsize=(20,9))
plt.plot(Ytesting , 'blue', linewidth=5)
plt.plot(predict,'r' , linewidth=4)
plt.legend(('Test','Predicted'))
plt.show()
```



```
[94] from sklearn.metrics import mean_squared_error
     mse = mean_squared_error(Ytesting, predict)
     print(f'MSE: {mse}')
```

```
MSE: 0.6607453763731668
```

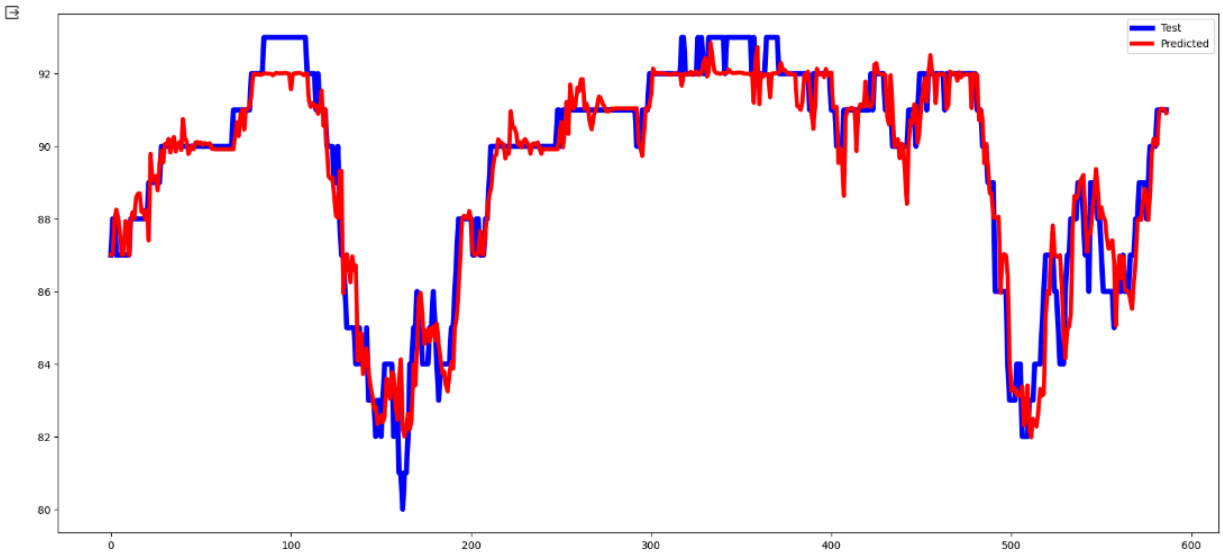
```
from sklearn.metrics import mean_absolute_error

# Mean Absolute Error (MAE)
mae = mean_absolute_error(Ytesting, predict)
print(f'MAE: {mae}')
```

```
MAE: 0.5581021674447003
```

```
[96] from sklearn.metrics import r2_score
     r2 = r2_score(Ytesting, predict)
     print(f'R-squared: {r2}')
```

```
R-squared: 0.9265861433825272
```

Humidity:




```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(Ytesting, predict)
print(f'MSE: {mse}')
```

```
MSE: 0.1256293580382873
```

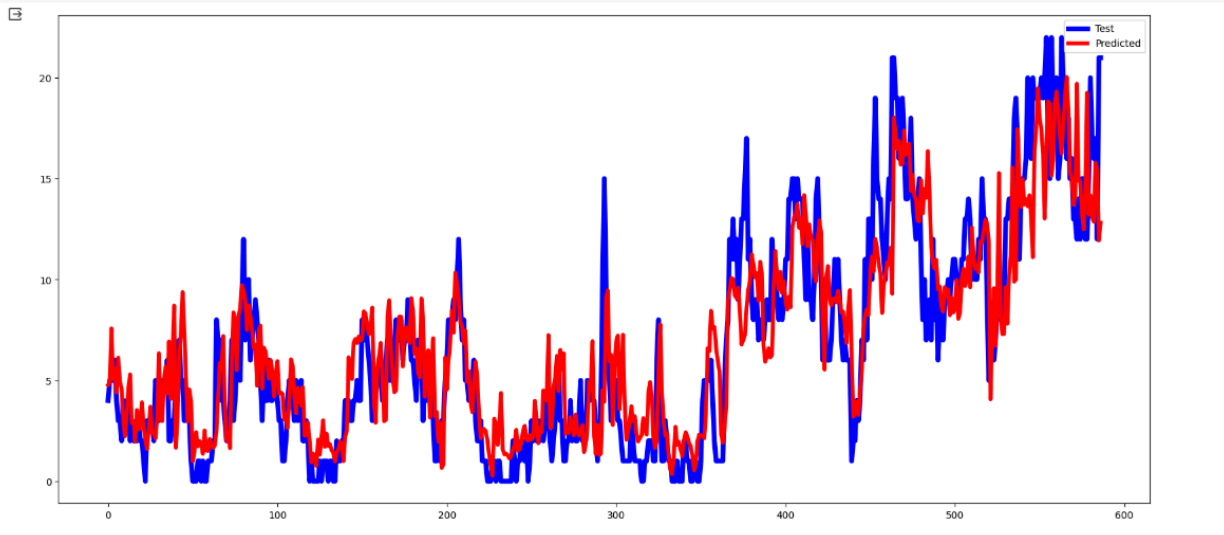
```
[55] from sklearn.metrics import mean_absolute_error

     # Mean Absolute Error (MAE)
     mae = mean_absolute_error(Ytesting, predict)
     print(f'MAE: {mae}')
```

```
MAE: 0.15425295074306924
```

```
[56] from sklearn.metrics import r2_score
     r2 = r2_score(Ytesting, predict)
     print(f'R-squared: {r2}')
```

```
R-squared: 0.8559324316241401
```

Wind Speed:




```
[135] from sklearn.metrics import mean_squared_error
      mse = mean_squared_error(Ytesting, predict)
      print(f'MSE: {mse}')
```

```
MSE: 7.14367861444042
```

```
from sklearn.metrics import mean_absolute_error

# Mean Absolute Error (MAE)
mae = mean_absolute_error(Ytesting, predict)
print(f'MAE: {mae}')
```

```
MAE: 2.0305014956342706
```

```
[137] from sklearn.metrics import r2_score
      r2 = r2_score(Ytesting, predict)
      print(f'R-squared: {r2}')
```

```
R-squared: 0.7578183516765923
```

Extracting TFLite Models:

```python
import tensorflow as tf

# Load or define your TensorFlow/Keras model
model = tf.keras.models.load_model('model_trained_wind.h5')

# Convert the model to TensorFlow Lite format
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS, tf.lite.OpsSet.SELECT_TF_OPS]
converter.experimental_new_converter = True  # Enable the new converter
converter._experimental_lower_tensor_list_ops = False  # Disable lowering tensor list ops
tflite_model = converter.convert()

#  Save the TensorFlow Lite model to a .tflite file
with open('converted_wind_model.tflite', 'wb') as f:
    f.write(tflite_model)
```

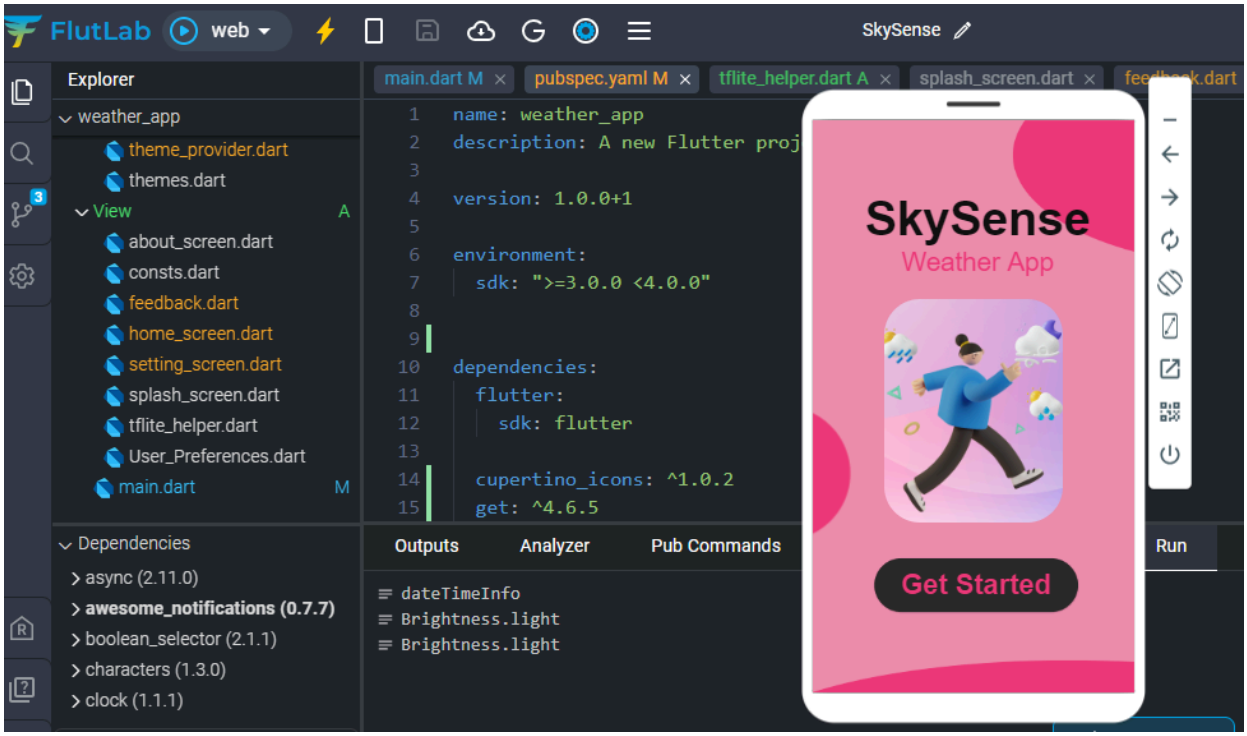
```
WARNING:tensorflow:Layer lstm_11 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_11 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
```

Integrating Flutter with TFLite:

```dart
import 'package:tflite/tflite.dart';

class TFLiteHelper {
  static Future<String?> loadModel() async {
    String? result = await Tflite.loadModel(
      model: "assets/model.tflite",
    );
    return result;
  }

  static Future<List?> runModelOnImage(String imagePath) async {
    var recognitions = await Tflite.runModelOnImage(
        path: imagePath, // required
        imageMean: 0.0, // defaults to 117.0
        imageStd: 255.0, // defaults to 1.0
        numResults: 5, // defaults to 5
        threshold: 0.2, // defaults to 0.1
        asynch: true // defaults to true
        );
    return recognitions;
  }
```

```yaml
version: 1.0.0+1

environment:
  sdk: ">=3.0.0 <4.0.0"


dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2
  get: ^4.6.5
  http: ^1.1.0
  intl: ^0.18.1
  weather: ^3.1.1
  provider: ^5.0.0
  awesome_notifications: ^0.7.4+1
  flutter_local_notifications: ^5.0.0+4
  wiredash: ^2.1.0
  shared_preferences: ^2.0.18
  tflite: ^1.1.2
```

Flutter Code:



main.dart:

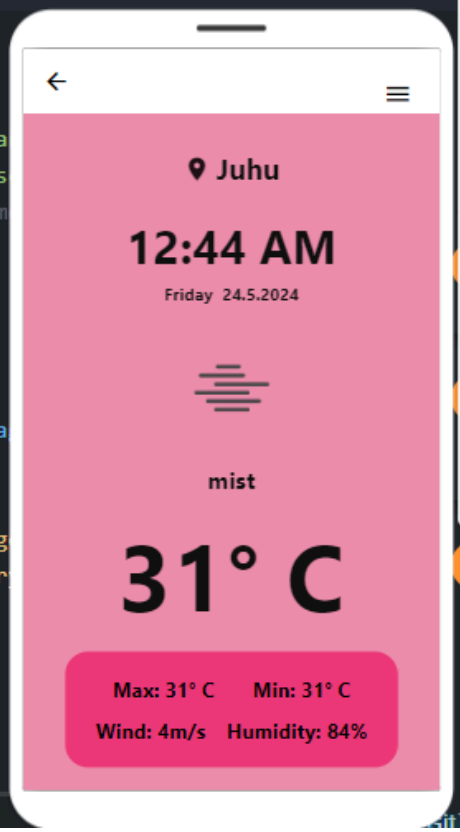
main.dart M × | pubspec.yaml M × | tflite_helper.dart A × | splash_screen.dart × | feedback.d

```dart
import 'package:awesome_notifications/awesome_notifications.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:weather_app/View/splash_screen.dart';
import 'package:weather_app/themes/theme_provider.dart';
import 'package:wiredash/wiredash.dart';
import 'package:weather_app/View/tflite_helper.dart';

void main() {
  AwesomeNotifications().initialize(
    null,
    [
      NotificationChannel(
        channelKey: "basic_channel",
        channelName: "SkySense",
        channelDescription: "Notifications on",
      ),
    ],
    debug: true,
  );
  runApp(ChangeNotifierProvider(
    create: (context) => ThemeProvider(),
```

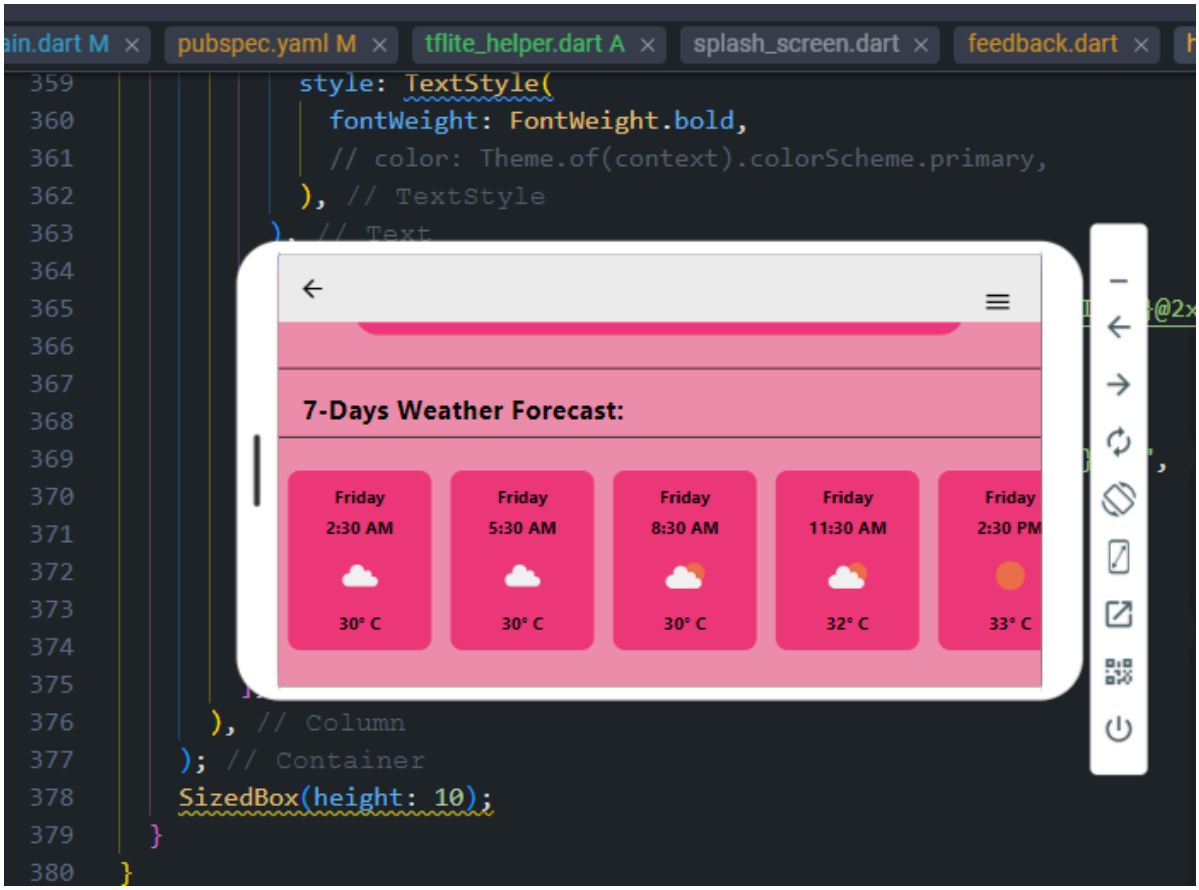
```dart
@override
Widget build(BuildContext context) {
  return Wiredash(
      projectId: "skysense-8weie4o",
      secret: "O7oUikYmfjJYAR7hRz7wU7ko23Ei98Iv",
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'Flutter Demo',
        theme: Provider.of<ThemeProvider>(context).themeData,
        // themeMode: notifier.isDark ? ThemeMode.dark : ThemeMode.li
        // darkTheme: darkTheme,
        // theme: ThemeData(
        //   colorScheme: ColorScheme.fromSeed(seedColor: Colors.de
        //   useMaterial3: true,
        // ),
        home: SplashScreen(),
      )); // MaterialApp // Wiredash
}
}
```

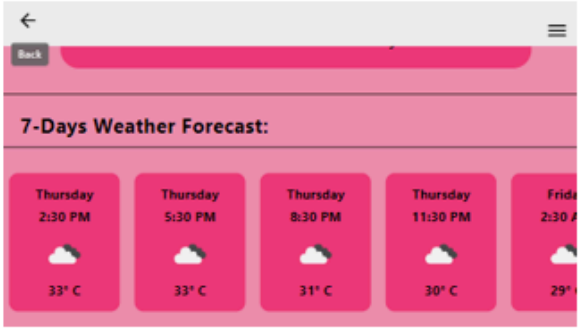
Home screen:

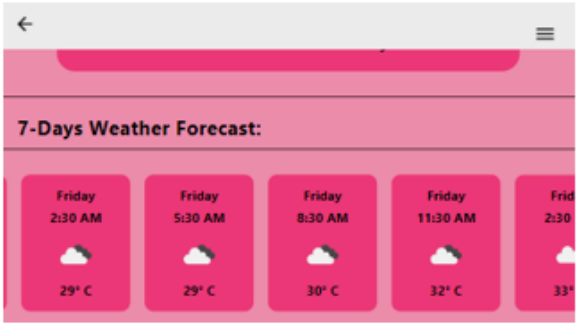
```
1  import 'dart:ui';
2  import 'package:flutter/material.dart';
3  import 'package:intl/intl.dart';
4  import 'package:weather/weather.dart';
5  import 'package:weather_app/View/consts.da
6  import 'package:weather_app/View/setting_s
7  // import 'package:weather_app/themes/them
8
9  class HomePage extends StatefulWidget {
10   const HomePage({Key? key});
11
12   @override
13   State<HomePage> createState() => _HomePa
14 }
15
16 class _HomePageState extends State<HomePag
17   final WeatherFactory _wf = WeatherFactor
18
19   Weather? _weather;
20   List<Weather>? _forecast;
21
22   @override
```

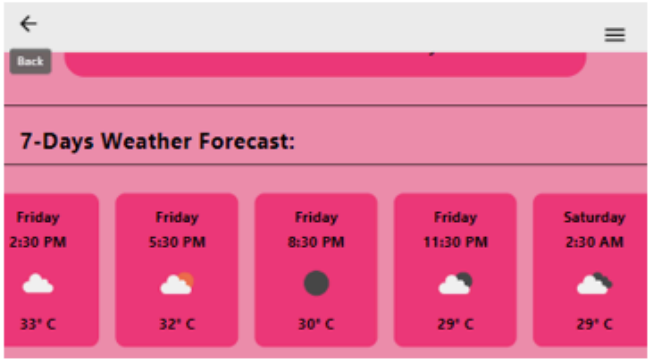


Outputs Analyzer Pub Commands Tests


ain.dart M
pubspec.yaml M
tflite_helper.dart A
splash_screen.dart
feedback.dart
style: TextStyle(
fontWeight: FontWeight.bold,
// color: Theme.of(context).colorScheme.primary,
), // TextStyle
), // Text
7-Days Weather Forecast:
Friday 2:30 AM 30° C
Friday 5:30 AM 30° C
Friday 8:30 AM 30° C
Friday 11:30 AM 32° C
Friday 2:30 PM 33° C
), // Column
); // Container
SizedBox(height: 10);
}
}

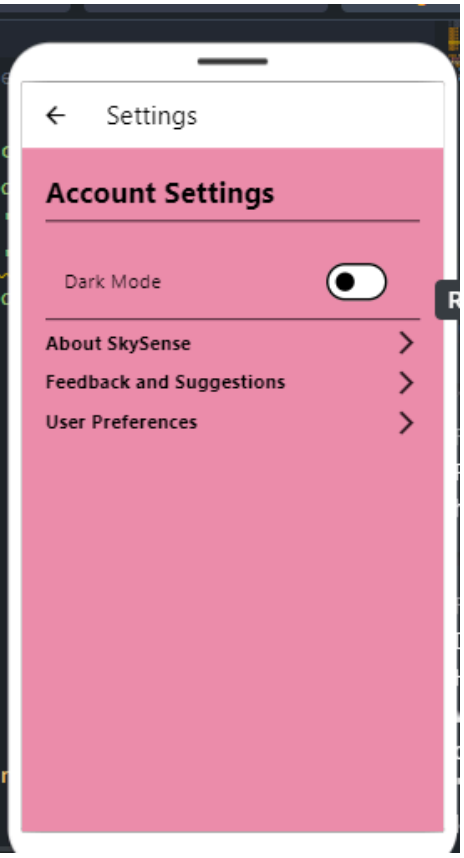

Back
7-Days Weather Forecast:
Thursday 2:30 PM 33° C
Thursday 5:30 PM 33° C
Thursday 8:30 PM 31° C
Thursday 11:30 PM 30° C


7-Days Weather Forecast:
Friday 2:30 AM 29° C
Friday 5:30 AM 29° C
Friday 8:30 AM 30° C
Friday 11:30 AM 32° C

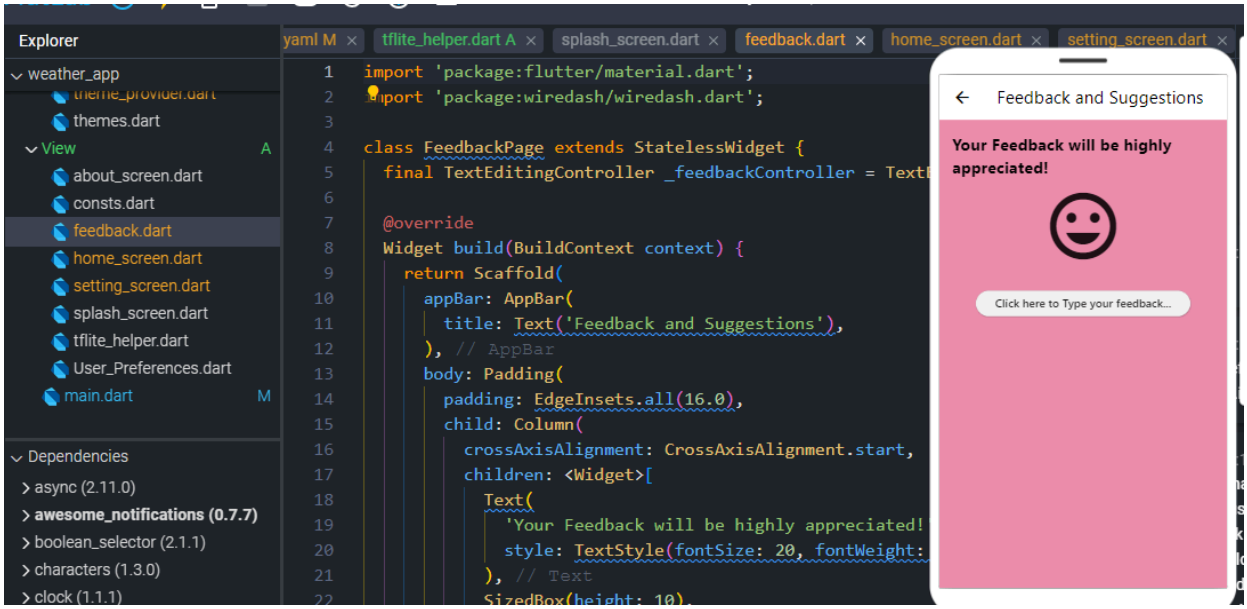

Back
7-Days Weather Forecast:
Friday 2:30 PM 33° C
Friday 5:30 PM 32° C
Friday 8:30 PM 30° C
Friday 11:30 PM 29° C
Saturday 2:30 AM 29° C

Settings Page:

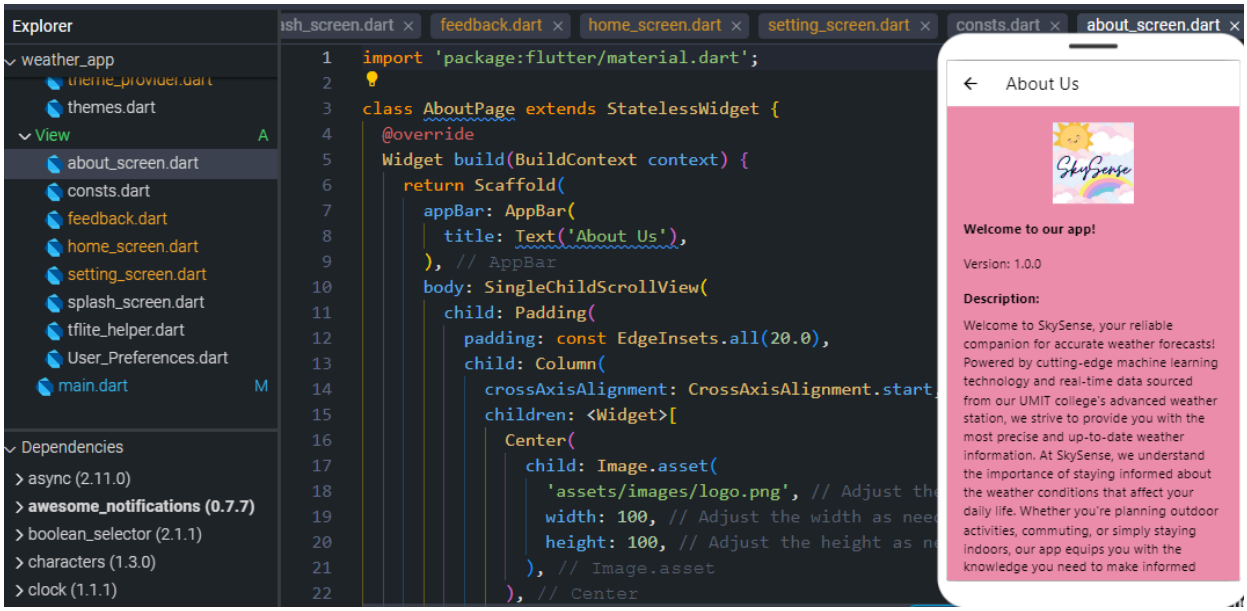
```dart
// import 'package:flutter/material.dart';
// import 'package:weather_app/View/about_scree
import 'package:flutter/material.dart';
import 'package:weather_app/View/User_Preferenc
import 'package:weather_app/View/about_screen.d
import 'package:weather_app/View/feedback.dart'
import 'package:weather_app/themes/themes.dart'
import 'package:weather_app/themes/theme_provid
import 'package:provider/provider.dart';

class SettingsPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Settings'),
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignmer
          children: <Widget>[
```



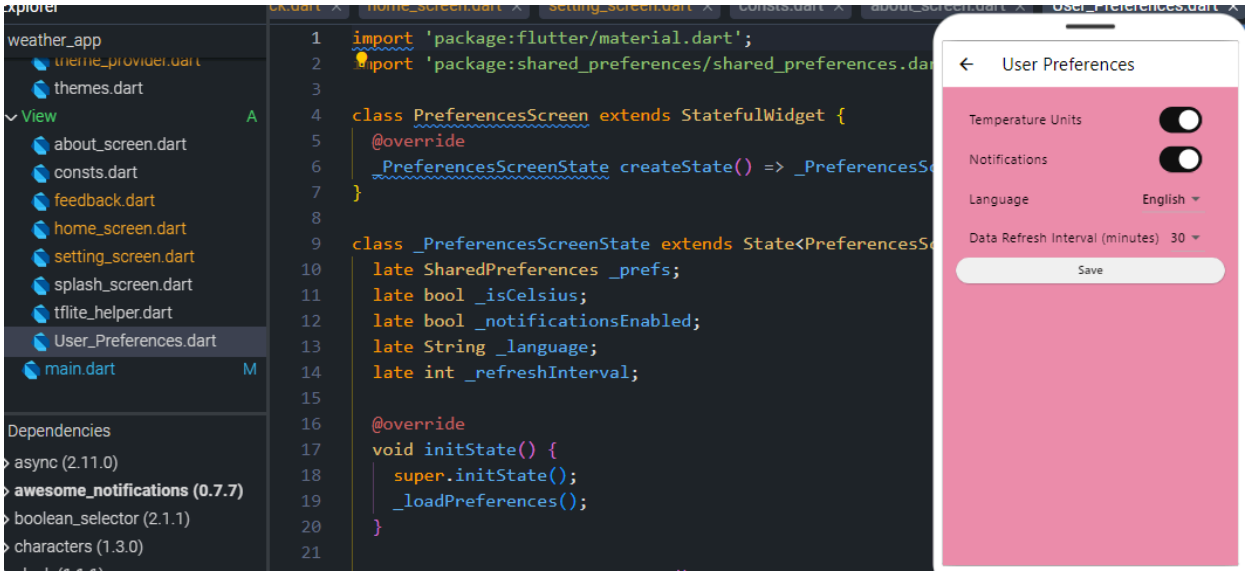
Feedback Page:



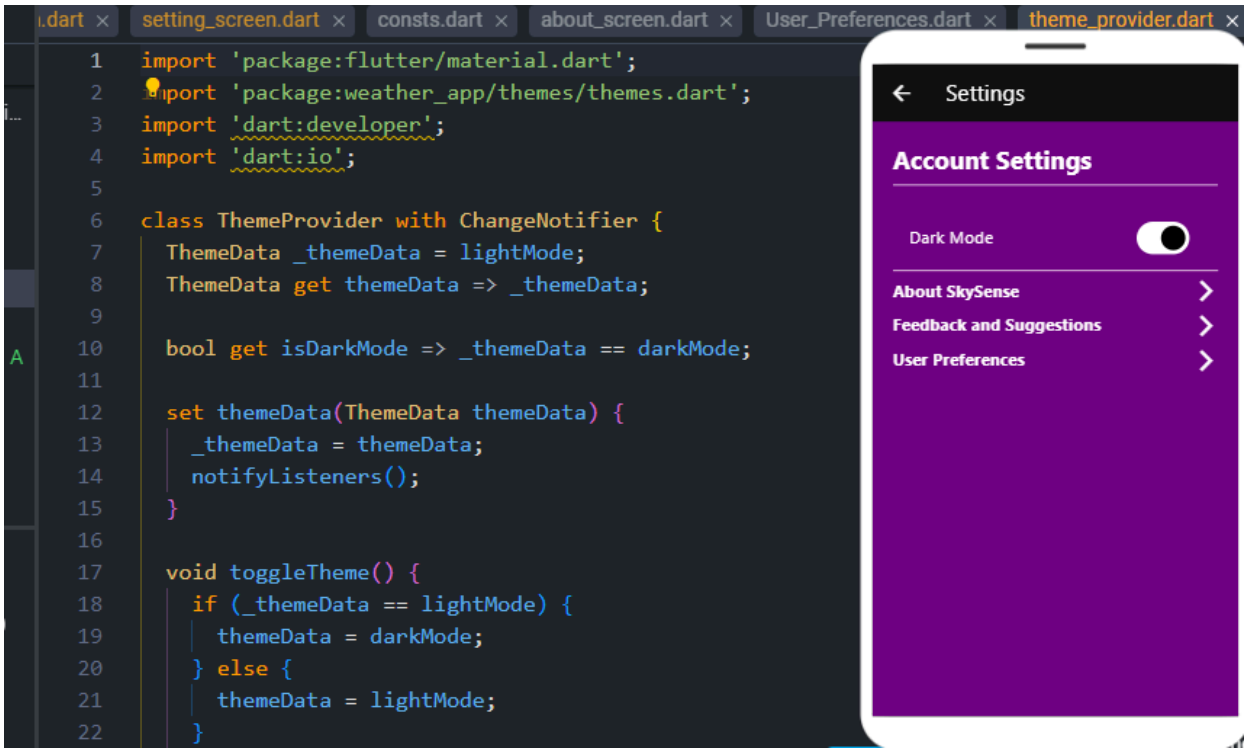
About Page:



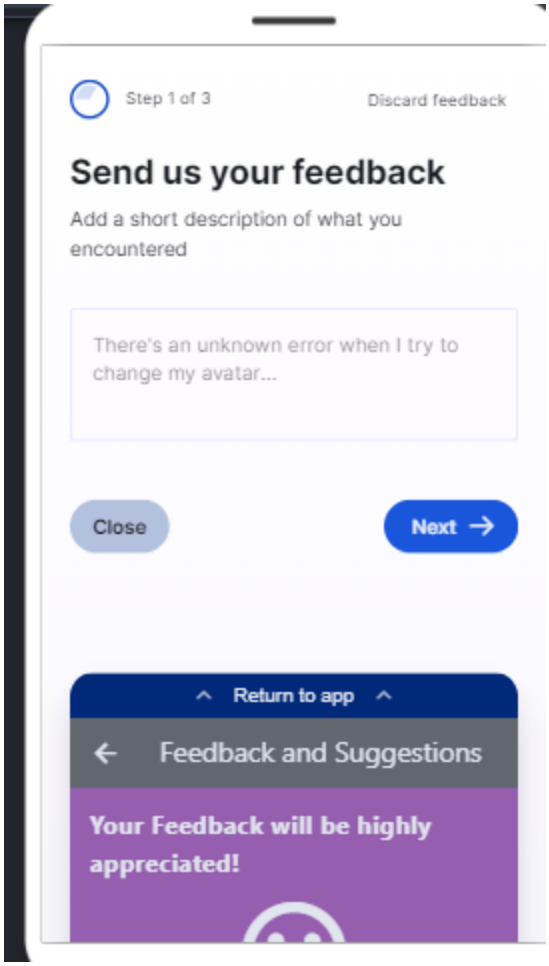
User Preferences Page:

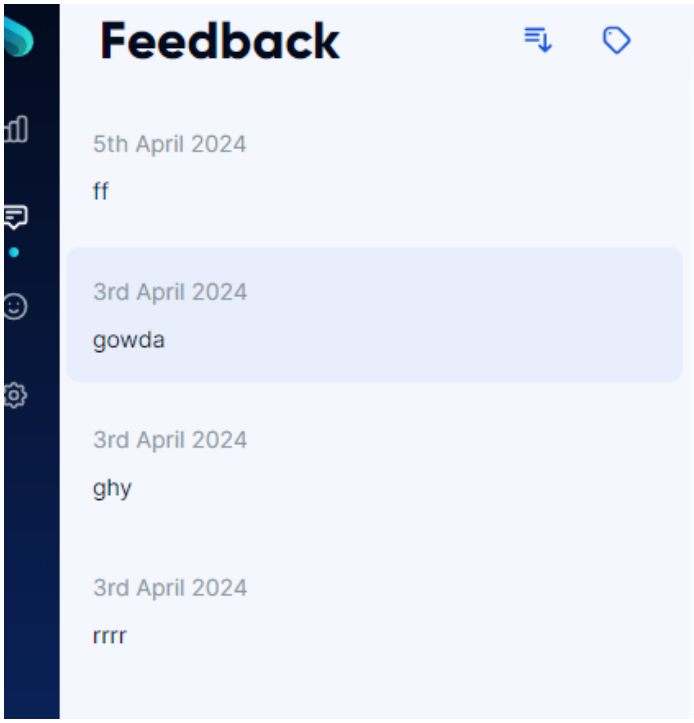


Light/ Dark Mode:



FeedBack Form Connected with wiredash for database collection:


Step 1 of 3
Discard feedback
Send us your feedback
Add a short description of what you encountered
There's an unknown error when I try to change my avatar...
Close
Next
Return to app
Feedback and Suggestions
Your Feedback will be highly appreciated!


Feedback
5th April 2024
ff
3rd April 2024
gowda
3rd April 2024
ghy
3rd April 2024
rrrr